Amendments to the Drawings

The attached sheet of drawings 9/11 includes changes to Fig. 26. Applicants have prepared and hereby submit Replacement Sheet 9/11. This sheet 9/11 replaces previously filed sheet 9/11.

Attachment: Replacement Sheet 9/11
Annotated Sheet showing changes.

REMARKS

Applicants have carefully reviewed the Office Action dated April 19, 2005. Claims 3-20 are pending in this application. Applicants have amended the specification and Fig. 26 and Claims 1, 5, 12 and 14 to more clearly point out the present inventive concept. Reconsideration and favorable action is respectfully requested.

Applicants have made some amendments to the Specification to clear up some minor changes that are noted with respect to the numbering and the such. These are relatively straightforward and should present no problem and should add no new matter. Applicants respectfully request entry of these amendments to both the Specification and the drawings.

Claims 3-20 stand rejected under 35 U.S.C. §102(e) as being anticipated by *Stroyan*, U.S. Patent No. 6,429,877. This rejections is respectfully traversed.

Applicants note with appreciation the Examiner's detailed response to Applicants' prior arguments and the questions that the Examiner has set forth. Applicants believe that the point of issue has clearly been set forth by the Examiner in the form of the question wherein the Examiner has requested Applicants to determine upon what basis does the *Stroyan* reference create an anti-aliasing value, if it is not a function of the vertices of the primitives, i.e., the polygon.

With respect to the *Stroyan* reference, Applicants believe that following description with the accompanying Claim Chart, will clarify the issue as to the portion of the claim wherein the rendering of the Z_s value is determined upon whether the rendering of a polygon is in the foreground or the background with respect to an edge pixel.

Stroyan defines on Col. 2, beginning at line 7 the following:

> Each pixel is represented with a center point, and an outer, circular rim. As illustrated, the number of pixels (illustrated in dash line) have at least some portion that intersects the edge 16. These are also referred to as edge pixels.

This can clearly be seen in Fig. 1b where the edge pixels in phantom outline, or the ones labeled, are edge pixels 20, 22, 18 and 24. Of these pixels, the center point of pixels 20, 22 and 24 are within the blue primitive (12). A pixel 18 and two pixels on opposite sides thereof are edge pixels but they have the sampling point outside of the blue triangle. This is an important distinction, as the above noted definition of edge pixels includes all of the pixels with the dotted line. However, Applicants will note that the edge pixel 22, although being described as being within the red primitive, has a sampling point that, as drawn, is actually inside the blue primitive. For discussion purposes, each of the edge pixels that has a sampling point outside of the primitive to which an edge is referred, are described as "outlander" edge pixels. It is important to understand how *Stroyan* works and the difference between the advantage provided by Applicants' present inventive concept, as defined by the claims, and the way *Stroyan* actually works under different rendering orders. The primary question is how *Stroyan* determines the coverage area for these edge pixels.

To further explore how the coverage for these pixels is determined from the standpoint of creating the value in the byte (162), this value will be referred to as an anti-aliasing (AA) value. First, a review of the flow chart of Fig. 6 in *Stroyan* may be pertinent. In this flow chart, there are two paths to take. The first path is when a pixel is determined as an edge pixel. The second path is the path taken when there the pixel is determined not to be an edge pixel. (It is noted that the term "edge pixel" is relative to the primitive being rendered. Thus, a pixel can be an edge pixel during the rendering of one primitive and the same pixel may not be an edge pixel during the rendering of another primitive). If there is a determination that a pixel is not an edge pixel, what occurs according to the flow chart is that the background color of the pixel is written to the frame buffer and then the coverage value or AA value is written to the byte (162) as a value of "11111." If it is an edge pixel, Applicants believe that the decisions made in the branch associated with the "YES" path are somewhat unclear. The first step in block (210) is to determine the coverage in the particular edge pixel and then to determine the direction

to the edge, in the block (212). However, the specification when dealing with an edge pixel with respect to this path in Fig. 6 is described beginning at Col. 8, line 64 as follows:

> . . . If a given pixel is determined to be an edge pixel, then the method may determine the coverage area for the current pixel (step 210). In this regard, the coverage area is the percentage of the pixel (containing the center point) that lies within the edge of the primitive.

From this description, one could read that edge pixels are only edge pixels having the center point thereof lying within the edge of the primitive. Applicants believe that this is correct only for the edge pixels 20, 22 and 24. However, this language does not specifically determine how outlander pixels are to be treated, since their center points do not lie within the primitive. Applicants believe that this is incorrect for all edge pixels. Certainly, for pixels inside the primitive, the language in Col. 8, beginning at line 64 is correct. However, for outlander pixels, the coverage area is more correctly determined as being the percentage of the pixel on the same side of the edge (16) as the center point of the edge pixel. Further, Fig. 6 fails to determine how to deal with the source color, as was done along the opposite path in block (204). In Col. 3, beginning at line 47, the following is set forth with respect to outlander pixels:

> In a conventional rasterization process, only the pixels having centerpoints within a primitive are covered by the primitive, and therefore rendered (see FIG. 1C). However, and in accordance with present invention, the region of interest is effectively extended to pixels that touch the primitive edge in any amount. This allows the invention to represent minority coverage of a pixel by an appropriate blending by a coverage percentage.
>
> One method of representing the primitive coverage of pixels with less than a 50% coverage value (e.g. the centerpoint is not covered) is to not modify the color buffer for this pixel, but rather only modify anti-alias (A/A) blend information to point toward the adjacent pixel having a centerpoint that is covered by the current primitive. This allows the minority contribution of this primitive to be considered while preserving original color value that represents the majority of this pixel's cover.

It can be seen, therefore, than an outlander edge pixel certainly comprises an edge pixel for the purpose of writing the AA value to the byte (162). During the rendering of a primitive, such as the blue primitive, and determining the AA value for an outlander edge pixel, the AA value will be stored but the color buffer will not be updated, i.e, the source color will remain the same. Thus, Applicants believe that Fig. 6 is not complete in the flow chart along a "YES" path.

The purpose for discussing this is to point out the fact that edge pixels are defined with respect to an edge of a primitive being rendered and only with respect to that primitive. If the same pixel is being evaluated during th rendering of another primitive, such as the red primitive associated with the red cube (10), there is no consideration given to the edge (16), as the rendering process does not realize that the edge (16) exists relative to another primitive in the system (current systems do not have a level of complexity to determine all edges that might exist within their space; rather, the rendering process merely considers only edges on the boundaries of the vertices for that primitive). Thus, the edge pixels 20, 22, 18 and 24 set forth in Fig. 1b are not edge pixels for the purpose of rendering the red primitive. As will be discussed hereinbelow, when the order of the rendering of the primitives is reversed, this will show the benefits of Applicants' present inventive concept and hopefully provide an answer to the Examiner's question as to "based on what does the reference *Stroyan* create an anti-aliasing value, if it is not a function of the vertices of the primitives (i.e. polygon)?" Further, it is also noted that the order in which polygons or primitives are rendered is not necessarily back to front. They can be rendered in any order. Although the order is not clearly set forth in *Stroyan*, Applicants will illustrate that the only way that the example set forth by *Stroyan* could possibly be interpreted is where the red primitive were rendered first and then the blue primitive. Applicants have provided color coded charts for the Examiner's reference and the attached Figs. 1-6. These figures depict the following:

> Figure 1 illustrates red-blue ordering and the rendering of the red primitive;
>
> Figure 2 illustrates red-blue ordering and the next step of rendering the blue primitive;

Figure 3 illustrates red-blue ordering and the step of determining the AA values;

Figure 4 illustrates blue-red ordering and the first step of rendering the blue primitive;

Figure 5 illustrates the step of determining the AA values during the rendering of the blue primitive;

Figure 6 illustrates blue-red ordering and rendering of the red triangle.

With specific reference to Figs. 1-3, this is believed to be the order in which *Stroyan* must operate.

Referring now to Fig. 1, the pixels in the area (14) are rendered and they are considered to be completed within the red cube (10) and, thus, the background color of all of the pixels will be set to red and the AA value will be set to "11111." None of the pixels are edge pixels, as the line (16) does not exist at the time of rendering of the red primitive. This is important when ordering is reversed, as will be described hereinbelow. Again, the red primitive is the first that is rendered in this example.

Referring now to Fig. 2, the next step will occur wherein the blue primitive is rendered. In this first step, all of the pixels having the sampling point thereof within the blue primitive will be considered to belong to that primitive and the background color will be set to blue. Note, again, that Applicants consider pixel (22) to be within the blue primitive. This figure essentially looks like Fig. 1c which correctly shows pixel (22) within the blue primitive.

Referring now to Fig. 3, there is illustrated the step of determining the AA value for all edge pixels. These are represented in phantom line pixels. As set forth in Fig. 1b, there are a total of five edge pixels, of which three are outlander pixels. Only one of those outlander pixels, pixel (18), being labeled. The background color of these outlander pixels is red but the coverage mask or AA value is determined based upon the edge (16) that did not exist during the rendering operation of the red primitive, such as pixel (17). As such, the AA value which was written to "11111" during the rendering

of the red primitive will be overwritten with the new value as a result of the rendering of the blue primitive and a determination that the outlander pixel is to have the AA value determined during rendering of the blue primitive. This will provide the correct blending operation and the correct smoothing operation in accordance with the *Stroyan* description.

The problem that Applicants have addressed in the current invention, as set forth in the claims, addresses what occurs when other primitives are rendered after the AA values for the outlander pixels are determined, and which contain the outlander pixels. Depending upon the order of the rendering, it is possible that the outlander pixels will be incorrectly overwritten.

During the rendering process for any primitive, the manner by which it captures the background of the pixel is by making the determination if the vertices of the center point of the pixel are the coordinates that are closest to the display, i.e., the foremost coordinates, relative to all of the previously rendered primitives. Thus, a depth value, Z-value, is determined and stored in a buffer, such that the only way that the background or source color of the pixel will be changed is if another primitive is rendered and the coordinates of the center point in that primitive are closer to the display. In Applicants' disclosure, this is set forth as the primary Z-value, Z_p. Thus, a Z_p value will be stored for each pixel in association with the color such that all that is necessary during the rendering of another primitive for that pixel is to determine if the coordinates of that pixel within that primitive have a shallower depth. If so, then the source color will be overwritten and the Z_p value will be overwritten.

To reverse the order of rendering as rendering the blue primitive first and then the red primitive, reference is made to Figs. 4-6. In Fig. 4, the first step is to render the blue primitive. This will result in the same operation as above wherein each pixel that is determined to have the center point thereof within the blue primitive will have the source color thereof changed to blue and the Z_p-value changed to the depth of that pixel in the Z-buffer. Of course, this assumes that the blue primitive is in the foreground which, for this example, is the case.

Referring now to Fig. 5, the next step will be to determine what pixels are edge pixels relative to the blue primitive. This is done similar to that described herein above wherein six pixels are determined to be edge pixels during rendering of the blue primitive and, of those six pixels, three are outlander pixels. It is noted that these outlander pixels have not, in this example, been found to exist within any primitive yet rendered and, therefore, that their Z_p-values would be maximum, i.e, at the maximum depth, and their AA value would be set to "11111." Once an outlander pixel is determined to be an edge pixel during the rendering of the blue primitive, such as pixel (18), then this pixel, although the Z_p- value is recognized as being the maximum depth and the color is not changed to the source color of the blue primitive, the AA value and the byte (162) would be changed for that outlander pixel. Again, this is correct for the edge (16). The problem exists in the next figure, Fig. 6. In this figure, the next step is to render the red primitive. Since the red primitive is beneath the blue primitive, the edge (16) will not exist and all of the blue pixels defined in Fig. 4 will remain with the source color of blue and the Z_p- value remaining the same. The reason for this is that the red primitive is beneath the blue primitive. However, for all pixels that are within the red primitive and outside of the edge (16) and the blue primitive, the source value will be changed and the Z_p value will be changed. However, note that with respect to Fig. 6, block (206), it is noted that whenever a pixel is not an edge pixel and pixel (18) is certainly not an edge pixel with respect to an edge in the red primitive, then the AA value will be written to "11111." This essentially overwrites the value determined during the rendering of the blue primitive. Thus, all the outlander pixels which were carefully considered with respect to the rendering of the blue primitive will be overwritten and will no longer be considered edge pixels. As such, the blending operation of the pixels along the edge with this ordering will be different than with the ordering of Figs. 1-3, and that in accordance with the description of the primary embodiment of *Stroyan*. *Stroyan* does not set forth any method to deal with this. The reason is that *Stroyan* does not consider what happens in this ordering and it certainly causes the "smoothing" operation to be distinctly different between the example of Fig. 6 and the example of Fig. 3, since half of the edge pixels have been overwritten.

What Applicants have done is to generate a Z_s-value which determines the depth of the AA mask for an edge pixel during the rendering of the blue primitive, the foremost primitive, and to retain that

value when an edge pixel is an outlander pixel in a deeper primitive. The Z_p-value is compared with the Z_s-value of the AA mask during rendering to determine if it had been written by the operation of rendering another primitive that was in the foreground. Thus, the AA value determined during rendering of the blue primitive for all outlander pixels would be retained during rendering of the red primitive if there were some information to determine that the outlander pixel had an existing AA value that was written in association with the rendering of a primitive that was closer to the display. Thus, as set forth in Applicants' Claim 3, the anti-aliasing value is retained as a function of the polygon being in the foreground during the rendering of a pixel in the polygon. As set forth in the claims, if the polygon that was associated with the definition of the AA mask is closer to the display than the rendering of other polygons containing the edge pixel such that outlander edge pixels cannot be overwritten. The only way that any decision can be made as to whether to retain a value is to have some type of depth value associated therewith. *Stroyan* does not disclose such.

Applicants believe that *Stroyan* is deficient in that it does not address how to maintain the AA value for outlander pixels when the rendering order is reversed. Apparently, that situation results in overwriting of the AA value and destroying the anti-aliasing operation. Thus, Applicants believe that this deficiency is the reason that *Stroyan* fails as a reference and, therefore, Applicants believe that *Stroyan* does not anticipate or obviate Applicants' present inventive concept, as defined by amended Claims 3-20 and, therefore, respectfully requests withdrawal of the 35 U.S.C. §102(e) rejection with respect to *Stroyan*.

Applicants have now made an earnest attempt in order to place this case in condition for allowance. For the reasons stated above, Applicants respectfully request full allowance of the claims as amended. Please charge any additional fees or deficiencies in fees or credit any overpayment to Deposit Account No. 20-0780/BBOY-25,415 of HOWISON & ARNOTT, L.L.P.

Respectfully submitted,
HOWISON & ARNOTT, L.L.P.
Attorneys for Applicants

Gregory M. Howison
Registration No. 30,646

GMH/yoc/keb

P.O. Box 741715
Dallas, Texas 75374-1715
Tel: 972-479-0462
Fax: 972-479-0464
August 18, 2005

REPLACEMENT SHEET

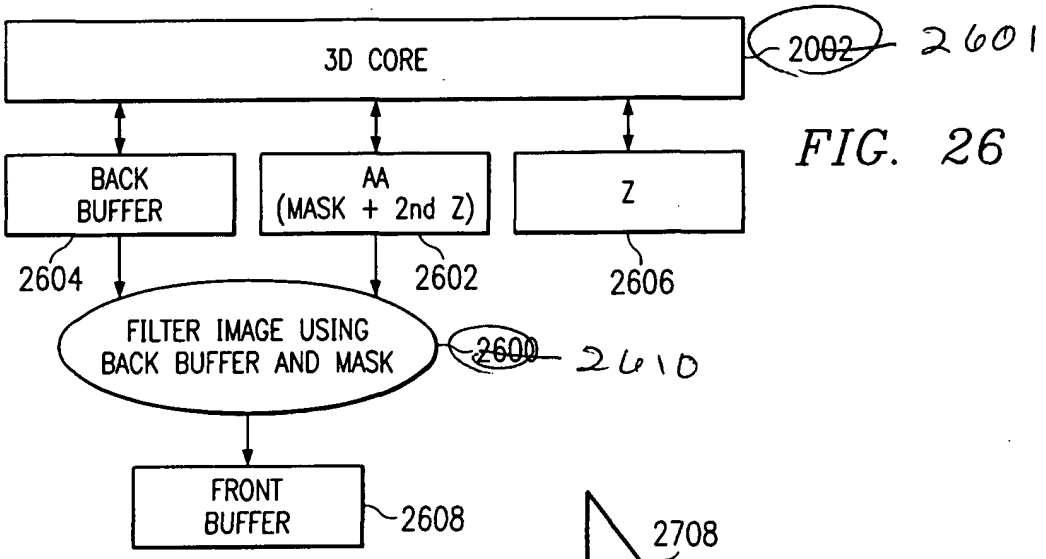

FIG. 26

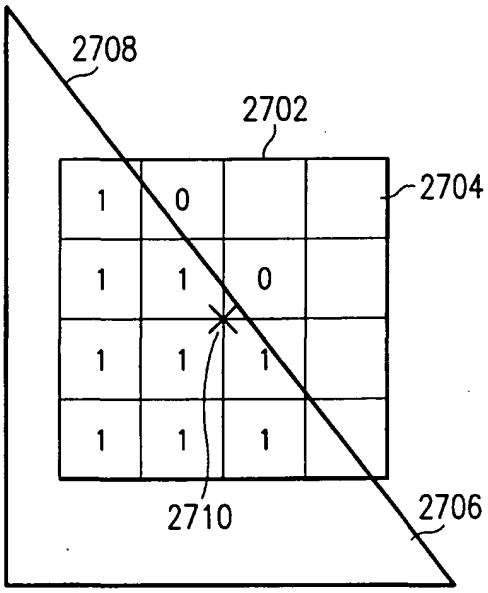

FIG. 27

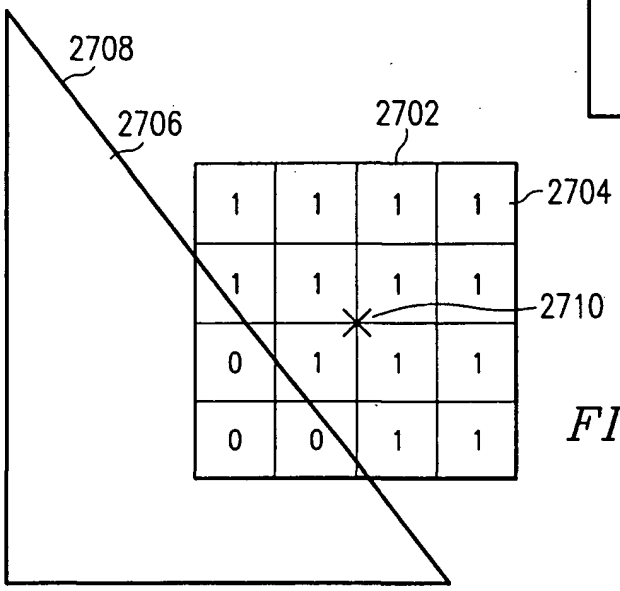

FIG. 28